Amendments to Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of the Claims:

1. (Currently Amended) A method of instantiating objects by a virtual machine, said method comprising:

receiving a first sequence of bytecodes to be executed by said virtual machine, wherein said first sequence of bytecodes is selected from a defined first set of executable virtual machine instructions implemented to conform with a virtual machine specification that includes said defined first set of executable virtual machine instructions;

selecting, at load time, a first-reduced instruction from a reduced-set of virtual machine instructions, wherein said first-reduced instruction represents two or more different virtual machine instructions in said first sequences of virtual machine instructions that were selected from said defined first set of executable virtual machine instructions, the virtual machine being arranged to execute the reduced-set of virtual machine instructions that provide substantially all of the functionality provided by said defined first set of virtual machine instructions, and wherein every one of the instructions in said defined first set of virtual machine instructions can be represented by at least one of the virtual machine instructions in the reduced-set of virtual machine instructions, and wherein said reduced-set of virtual machine instructions consists of a number of virtual machine instructions which is less than the number executable virtual machine instructions in said defined first set of virtual machine instructions;

translating, at load time, said two or more different virtual machine instructions in said first sequence into said first-reduced instruction from said reduced-set of virtual machine instructions;

~~generating~~ determining, after said translating, a second sequence of bytecodes that includes said first-reduced instruction, thereby representing said first sequence of bytecodes with a second sequence which includes at least ~~one~~ said first-reduced instruction, selected from said reduced-set of virtual machine instruction, that replaces said two or more different virtual machine instructions in said first sequence;

determining, at load time, whether said second sequence of ~~bytcodes~~ bytecodes includes an instantiation instruction immediately followed by a duplicate stack instruction;

~~generating~~ determining, at load time, a macro instruction that represents said instantiation instruction and said duplicate stack instruction that immediately follows said instantiation instruction;

loading in said virtual machine prior to execution time, said macro instruction instead of said instantiation instruction and said duplicate stack instruction; and

executing said macro instruction to instantiate a new object.

2. (Canceled)

3. (Currently Amended) A method as recited in claim 1, wherein said ~~Java~~ macro instruction is generated during a bytecode verification phase.

4. (Previously Presented) A method as recited in claim 1, wherein said virtual machine internally represents instructions as a pair of streams.

5. (Previously Presented) A method as recited in claim 4,

wherein said pair of streams includes a code stream and a data stream,

wherein said code stream is suitable for containing a code portion of said macro instruction, and

wherein said data stream is suitable for containing.

6. (Previously Presented) A method as recited in claim 5,

wherein said macro instruction is generated only when said virtual machine determines that said macro instruction should be generated.

7. (Previously Presented) A method as recited in claim 6, wherein said determination of whether said macro instruction should be generated is made based on a predetermined criteria.

8. (Previously Presented) A method as recited in claim 7, wherein said predetermined criteria is whether an instantiation instruction is immediately followed by a duplicate stack more than a predetermined number of times.

9-21 (Canceled)

22. (Currently Amended) A computer system for instantiating objects by a virtual machine, wherein said computer system ~~is capable of~~ operates to:

~~receiving~~ receive a first sequence of bytecodes to be executed by said virtual machine, wherein said first sequence of bytecodes is selected from a defined first set of executable virtual machine instructions implemented to conform with a virtual machine specification that includes said defined first set of executable virtual machine instructions;

~~selecting~~ select, at load time, a first-reduced instruction from a reduced-set of virtual machine instructions, wherein said first-reduced instruction represents two or more different virtual machine instructions in said first sequences of virtual machine instructions that were selected from said defined first set of executable virtual machine instructions, the virtual machine being arranged to execute the reduced-set of virtual machine instructions that provide substantially all of the functionality provided by said defined first set of virtual machine instructions, and wherein every one of the instructions in said defined first set of virtual machine instructions can be represented by at least one of the virtual machine instructions in the reduced-set of virtual machine instructions, and wherein said reduced-set of virtual machine instructions consists of a number of virtual machine instructions which is less than the number executable virtual machine instructions in said defined first set of virtual machine instructions;

~~translating~~ translate, at load time, said two or more different virtual machine instructions in said first sequence into said first-reduced instruction from said reduced-set of virtual machine instructions;

~~generating~~ determine, after said translating, a second sequence of bytecodes that includes said first-reduced instruction, thereby representing said first sequence of bytecodes with a second sequence which includes at least ~~one~~ said first-reduced instruction, selected from said reduced-set of virtual machine instruction, that replaces said two or more different virtual machine instructions in said first sequence;

~~determining~~ determine, at load time, whether said second sequence of ~~bytcodes~~ bytecodes includes an instantiation instruction immediately followed by a duplicate stack instruction;

~~generating~~ determine, at load time, a macro instruction that represents said instantiation instruction and said duplicate stack instruction that immediately follows said instantiation instruction;

~~loading~~ load in said virtual machine prior to execution time, said macro instruction instead of said instantiation instruction and said duplicate stack instruction; and

~~executing~~ execute said macro instruction to instantiate a new object.

23. (Previously Presented) A computer system as recited in claim 22, wherein said macro instruction is generated during a bytecode verification phase.

24. (Previously Presented) A computer system as recited in claim 22, wherein said virtual machine internally represents instructions as a pair of streams.

25. (Previously Presented) A computer system as recited in claim 24,

wherein said pair of streams includes a code stream and a data stream,

wherein said code stream is suitable for containing a code portion of said macro instruction, and

wherein said data stream is suitable for containing a data portion.

26. (Previously Presented) A computer system as recited in claim 22, wherein said macro instruction is generated only when said virtual machine determines that said macro instruction should be generated.

27. (Previously Presented) A computer system as recited in claim 26, wherein said determination of whether said macro instruction should be generated is made based on a predetermined criteria.

28. (Previously Presented) A computer system as recited in claim 27, wherein said predetermined criteria is whether an instantiation instruction is immediately followed by a duplicate stack instruction more than a predetermined number of times.

29. (Currently Amended) A computer readable medium including computer program code for instantiating objects by a virtual machine, comprising:

computer program code for receiving a first sequence of bytecodes to be executed by said virtual machine, wherein said first sequence of bytecodes is selected from a defined first set of executable virtual machine instructions implemented to conform with a virtual machine specification that includes said defined first set of executable virtual machine instructions;

computer program code for selecting, at load time, a first-reduced instruction from a reduced-set of virtual machine instructions, wherein said first-reduced instruction represents two or more different virtual machine instructions in said first sequences of virtual machine instructions that were selected from said defined first set of executable virtual machine instructions, the virtual machine being arranged to execute the reduced-set of virtual machine instructions that provide substantially all of the functionality provided by said defined first set of virtual machine instructions, and wherein every one of the instructions in said defined first set of virtual machine instructions can be represented by at least one of the virtual machine instructions in the reduced-set of virtual machine instructions, and wherein said reduced-set of virtual machine instructions consists of a number of virtual machine instructions which is less than the number executable virtual machine instructions in said defined first set of virtual machine instructions;

computer program code for translating, at load time, said two or more different virtual machine instructions in said first sequence into said first-reduced instruction from said reduced-set of virtual machine instructions;

computer program code for ~~generating~~ determining, after said translating, a second sequence of bytecodes that includes said first-reduced instruction, thereby representing said first sequence of bytecodes with a second sequence which includes at least ~~one~~ said first-reduced instruction, selected from said reduced-set of virtual machine instruction, that replaces said two or more different virtual machine instructions in said first sequence;

computer program code for determining, at load time, whether said second sequence of ~~bytcodes~~ bytecodes includes an instantiation instruction immediately followed by a duplicate stack instruction;

computer program code for ~~generating~~ determining, at load time, a macro instruction that represents said instantiation instruction and said duplicate stack instruction that immediately follows said instantiation instruction;

computer program code for loading in said virtual machine prior to execution time, said macro instruction instead of said instantiation instruction and said duplicate stack instruction; and

computer program code for executing said macro instruction to instantiate a new object.

30. (Previously Presented) A computer readable medium as recited in claim 29, wherein said macro instruction is generated during a bytecode verification phase.

31. (Previously Presented) A computer readable medium as recited in claim 29, wherein said virtual machine internally represents instructions as a pair of streams.

32. (Previously Presented) A computer readable medium as recited in claim 31,

wherein said pair of streams includes a code stream and a data stream,

wherein said code stream is suitable for containing a code portion of said macro instruction, and

wherein said data stream is suitable for containing data.

33. (Previously Presented) A computer readable medium as recited in claim 29, wherein said macro instruction is generated only when said virtual machine determines that said macro instruction should be generated.

34. (Previously Presented) A computer readable medium as recited in claim 33, wherein said determination of whether said macro instruction should be generated is made based on a predetermined criteria.

35. (Previously Presented) A computer readable medium as recited in claim 34, wherein said predetermined criteria is whether an instantiation instruction is immediately followed by a duplicate stack instruction more than a predetermined number of times.